云环境下副本优化放置策略研究 *

王　鑫[1,2]，孟　雨[1]，覃　琴[2]，蒋　华[1]

(1. 桂林电子科技大学 计算机与信息安全学院，广西 桂林 541004; 2. 桂林电子科技大学 海洋信息工程学院，广西 北海 536000)

摘　要：为了提高云计算数据调度和副本访问的效率，对副本策略中的副本放置问题进行研究，提出一种基于蚁群算法的副本放置策略。根据自然界中蚁群觅食的原理，把蚁群算法应用于副本放置的整个过程；利用信息素的动态更新以及拉普拉斯概率分布改进的蚁群算法，得出一组最优解进行副本放置。在 CloudSim 平台上进行了仿真模拟，实验结果表明，提出的方案在平均作业完成时间、网络利用率和负载均衡度上优于原始蚁群算法，并在一定程度上降低了副本放置时间消耗和网络负载。

关键词：云计算；蚁群算法；副本放置；负载均衡

中图分类号：TP391　　doi: 10.19734/j.issn.1001-3695.2018.10.0760

Research on replica optimal placement strategy in cloud environment

Wang Xin[1,2], Meng Yu[1], Qin Qin[2], Jiang Hua[1]

(1. *College of Computer Science & Engineering*, *Guilin University of Electronic Technology*, *Guilin Guangxi* 541004, *China*; 2. *School of Marine Information Engineering*, *Guilin University of Electronic Technology*, *Beihai Guangxi* 536000, *China*)

Abstract: In order to improve the efficiency of data scheduling and replica access in cloud computing, the paper studied the replica placement problem in replica strategy and proposed a replica placement strategy based on ant colony algorithm. According to the principle of ant colony foraging in nature, ant colony algorithm applies to the entire process of replica placement. The pheromone dynamic updating and Laplace probability distribution improves the ant colony algorithm, a set of optimal solutions is obtained for replica placement. Simulation is on the CloudSim platform. The experimental results show that the proposed scheme is superior to the original ant colony algorithm in average job completion time, network utilization and load balance, and reduces the copy placement time consumption and network load to a certain extent.

Key words: clouding computing; ant colony algorithm; replica placement; load balancing

0　引言

随着互联网的发展，复杂而庞大的数据需要被处理。据互联网数据中心的研究报告显示，未来数据的增长是几何式的，特别是移动终端。因此，海量数据的及时有效处理就成为了亟待解决的问题。云计算技术的兴起，尤其是云存储技术的出现，使多种类型的存储设备连接起来进行信息交互，并实现了协同合作。云计算环境由于数据量大和存储设备和网络设备差异的原因，会导致数据的丢失和出错，而副本技术的出现，能够有效的解决这种问题。通常在存储系统中采用多个副本，这样可以保证存储系统中数据的高可用性和高容错率。

由于系统性能和节点负载密切相关，云存储记录会表现新近数据的局部性特点。如果系统负载不均衡，单个节点的数据存储负载量会增大，用户对节点的访问性能将大大降低。同样，热点数据分布不均衡也会影响用户的读写性能。因此，如何在云存储系统中合理放置副本是一个值得研究的问题。

为了更加有效地改善云计算的副本技术性能，学者针对云计算中的副本策略所遇到的问题展开了研究。文献[1]针对云存储系统中节点的异构性，结合改进的粒子群算法对副本选择策略进行改进，使副本选择的效率得以提升，满足了用户的需求。文献[2]针对云计算中的副本问题，从副本选择方面对副本进行了优化，并与蚁群算法相结合。对虚拟机的利用状况进行了探究，最终达到优化了负载均衡。文献[3]考虑到数据开销和存储开销，提出兼顾成本和存储空间的副本策略。利用遗传算法和 Dijkstra 算法进行优化，取得了很好的效果。文献[4]为使副本能在存储节点上被合理分配，通过建立关系模型维持最小副本数量，提高数据副本可用性。文献[5]考虑到云存储系统中副本访问开销大的问题。提出把萤火虫算法应用于副本放置过程中，利用萤火虫的自然属性，建立相关的模型。仿真实验表明该算法对副本放置有着较好的效果。文献[6]提出异构环境下云计算副本动态管理模型，可以有效减少节点之间的数据传输。文献[7]提出动态副本分布方法，有效平衡了副本开销和访问性能之间的关系，优化了副本分布。文献[8]为改进 Qos 和操作开销，提出一种有效的贪婪启发式算法来解决副本放置问题，通过布局和细化，优化了计算的时间，取得了不错的效果。文献[9]综合考虑统计周期、文件大小、工作环境等多种因素，提出基于热度分析的副本创建方法，根据文件的访问热度，动态调整副本，提高了服务性能。文献[10]提出一种在云中动态数据复制方法，通过使用目录索引副本位置信息，最终将副本存储在有足够空间的系统中，以此来增加资源可用性和达到最小访问成本。

收稿日期：2018-10-17；**修回日期：**2018-11-26　**基金项目：**2016 广西高校中青年教师基础能力提升项目（ky2016YB150）

作者简介：王鑫（1976-），男，陕西蓝田人，副教授，硕士，主要研究方向为无线传感器网络、云计算技术；孟雨（1991-），男(通信作者)，山东淄博人，硕士研究生，主要研究方向为云计算技术（294519713@qq.com）；覃琴（1985-），女，广西桂林人，实验师，硕士，主要研究方向为无线传感器网络；蒋华（1963-），男，河南信阳人，教授，博士，主要研究方向为数据库系统、信息安全.

上述文献虽然对副本技术进行了相关的研究，但对云计算中的数据调度和访问效率方面的研究仍有所欠缺，且在副本管理技术中，作为关键问题的副本放置研究相对较少，还需进一步研究。本文在研究副本放置的特性、现有蚁群算法基础上，提出了基于改进蚁群算法的副本放置研究策略。该算法利用蚁群信息素动态更新和拉普拉斯概率分布改进的蚁群概率选择，最终形成本文中的改进蚁群算法。充分利用蚁群算法的分布性、随机性、反馈性特点，并结合副本放置，形成本文的解决方案。通过云计算仿真器 CloudSim 进行模拟，该算法在系统平均作业完成时间、网络利用率和负载均衡等方面有较好的表现。

1　副本放置问题描述

本文的研究在于副本放置问题，而副本管理中的其他问题并不完全在此囊括。在云存储系统中，在副本被创建之后，用户需要对副本进行合理的放置以保证数据的访问的性能。系统会在一系列副本中选出最佳的副本进行放置。不同系统的副本放置的标准也会不同。当有符合系统要求多个副本需要放置时，通过统计不同文件的访问开销，首先对文件访问频率贡献大的副本进行放置。定义文件的访问频率为

$$file_r(f)=\sum_{t=1}^{T}(2^{(t-T)}\times C_f^t),r=\{1,2,\ldots R\} \tag{1}$$

其中：R 表示整个系统中的副本的个数，$file_r(f)$ 表示高频率文件在副本 r 中频率的大小，用 C_f^t 来表示文件在第 t 个时间周期内被请求访问的次数。

文件所应放置的副本的数量 N_r^f 为

$$N_r^f=\left\lceil N\times\frac{file_r(f)}{\sum_{r=1}^{R}file_r(f)}\right\rceil,r=\{1,2,\ldots R\} \tag{2}$$

最后响应给请求的用户。副本放置中有以下两个关键的过程：

a) 根据用户请求中给定的检索文件名，通过云存储系统的副本管理器查找该检索文件名对应的若干个副本位置信息集合。

b) 在问题 a) 的基础上，查找出的若干个副本信息集合后，利用相关副本放置策略再进行相关副本的放置。

为了提高存储资源的利用率和数据文件的可靠性，应该综合考虑网络的访问性能、负载均衡、响应延迟和存储开销等方面。满足用户的动态需求。如放置副本时，需考虑是放置在本地还是远程放置。因此，可以说副本放置问题是综合多种因素进行求解优化的过程。

2　基于蚁群优化的副本放置算法

蚁群算法是通过模拟自然界中蚂蚁觅食行走的路径而产生的算法。蚂蚁在觅食过程中，会留下信息素进行群体的交流，信息素的多少决定了蚂蚁在此路径蚂蚁的多少。蚂蚁通过这种正反馈的形式使蚁群逐渐收敛，最终蚂蚁会在蚁窝和食物源中找到一条最短的路径。

2.1　蚁群优化算法核心描述

本节介绍蚁群算法在解决副本放置问题的基本思路，具体如下所示：

a) 把蚂蚁觅食时行走的路径抽象成目标放置前选择的过程，把路径集合抽象成一个解空间。

b) 蚂蚁移动过程中，会留下信息素。信息素会随着时间在较短路径上越来越浓，选择这条路的蚂蚁也会越来越多。

c) 蚁群信息素的正反馈作用，使蚁群的行走路径逐渐变得统一，最终到达目的地，完成对目标的放置。此条路径便是目标放置的最优解。

根据上面的思路，把蚁群抽象成选择放置副本的检索文件 RF(retrieve files)，同时会为该文件创建和寻找相应的副本。把蚂蚁行走路径的集合抽象为副本对象的集合 $R=\{r_1,r_2,\ldots r_n\}$，这样蚂蚁觅食的过程就变成了检索文件寻找相应副本并放置的过程。假设蚂蚁寻食行走的路径集合为 $P=\{p_1,p_2,\ldots,p_n\}$,这里 n 为蚂蚁的个数；觅食过程中所产生的信息素表示为集合 $\Gamma=\{\tau_1,\tau_2,\ldots\tau_n\}$。

设置整个蚁群是一个可行解的解空间，整个蚁群表示为 A，整个流程开始之前，本文需要对副本的信息素值进行初始化：

$$\tau_i(0)=\frac{readspeed}{replicasize} \tag{3}$$

其中：*replicasize* 指的是副本大小，*readspeed* 指的是读取的速度。从公式中可以看出读取速度越大，副本越小，信息素的初值就会越大。

当副本被访问多次之后，信息素会相应的发生变化，信息素相应的属性值也需要调整，表达式为

$$\tau_i(t+1)=\rho\cdot\tau_i(t)+\Delta\tau_i(t) \tag{4}$$

信息素挥发系数 ρ 的设定是否合理会影响蚁群的搜索能力和计算效率。在结合传统信息素更新方法上，引入动态改变 ρ 值的方法，使 ρ 能自适应的改变大小，从而保证算法的综合性能。现给出本文的调整策略。

$$\rho_a(t)=1-\ln(t)/\ln(t+c) \tag{5}$$

$$\rho(t)=\begin{cases}\rho_{min} & \rho_a(t)<\rho_{min}\\ \rho_a & \rho_{min}(t)\le\rho_a(t)\le\rho_{max}\\ \rho_{max} & \rho_a(t)\ge\rho_{max}\end{cases} \tag{6}$$

其中：c 为常数，ρ 的取值是信息素动态更新的关键，ρ 太大会使全局搜索能力下降，ρ 太小会使局部搜索能力变差，收敛的速度也会变慢。为了使挥发性系数 ρ 在一定范围内具有自适应性，挥发系数 ρ 控制在 $[\rho_{min},\rho_{max}]$，并对其取值为 [0.2,0.8]，这样可以拉开副本间信息素的差距，并在后期加快收敛的速度。

蚁群在路径上留下的信息素的浓度和副本能否被放置有着密切的联系，在信息素浓度越高的地方，副本被放置的概率就会越大。副本放置的概率公式定义如下：

$$P_j(t)=\begin{cases}\dfrac{[\tau_j(t)]^{\alpha}[\eta_j(t)]^{\beta}}{\sum_{u=1}^{n}[\tau_j(t)]^{\alpha}[\eta_j(t)]^{\beta}} & j,u\in\text{副本}\\ 0 & \text{其他}\end{cases} \tag{7}$$

其中：$\tau_j(t)$ 指的是副本 j 在 t 时刻的信息素的浓度，η_i 表示副本本身的固有属性。α 和 β 分别分别指代副本当前的信息素启发因子和期望启发因子。如果 α 比 β 的值大，说明在节点选择中，信息素浓度的作用比副本固有属性的影响大。为避免蚁群算法过早陷入局部最优，使算法综合求解性能较好，经过实验验证，本文取 α 为 3，β 为 1.5。

在传统的蚁群算法中，副本放置的概率选择都是采用随机概率 *rand* 结合式(7)进行概率匹配。随机性概率选择策略可以避免出现停滞现象，但是收敛的速度慢。为使算法能较快的收敛于最优解，在实验的基础上，本文引入了拉普拉斯概率分布的思想，采用最大概率选择的方法，在原有蚁群算法的基础上进行了改进。在副本放置的过程中，先计算出转移概率最大的对象 *MaxP*，再进行循环遍历，计算出副本放

置对象的概率和最大概率 *MaxP* 之间的距离，选取最靠近 *MaxP* 的副本进行放置，可得公式：

$$P(i)=[MaxP-rand, MaxP+rand], i=\{1,2,...m\} \quad (8)$$

则第 i 个副本对象将会被选择放置。

由式(7)可得副本信息素的浓度越高，则副本在被选择放置的时候，被选中的概率就会越大，但是在考虑放置副本的时候，节点的负载和网络的畅通也是考虑的重要的因素。本文引入类拉普拉斯概率分布选择最大概率副本进行放置的方法，可以有效防止网络的拥塞并在一定的程度上节省了存储的空间。

2.2 蚁群优化副本放置流程

云存储系统在受到副本文件 RF 的请求后，利用本文所提出的副本放置的方法对信息素的副本依次进行放置。下面是利用蚁群算法对副本文件进行放置的流程：

a)初始化副本信息素。

b)根据式(1)计算副本文件 RF 的访问频率和存储的节点情况。

c)若副本文件存储在远程节点，则依据改进蚁群算法进行副本的放置。

d)对副本对象进行概率匹配，据选择的概率根据式(7)(8)选出合适的副本。

e)选择出副本之后，数据传往终端，终端在获取副本之前，使用式(5)减小副本信息值，降低被重复访问的概率，平衡节点负载。

f)若副本在本地进行存储，则直接读取本地副本，不必执行蚁群副本放置。

g)若副本的相关数据没有被读取成功，则跳转 Step2 进行下一个副本对象的读取。

h)当副本被读取成功的时候，判断是不是最后一个副本对象，如果是，则结束副本放置，如果不是，再次读取下一个副本对象，进行改进蚁群副本放置。

其算法流程图如图 1 所示。

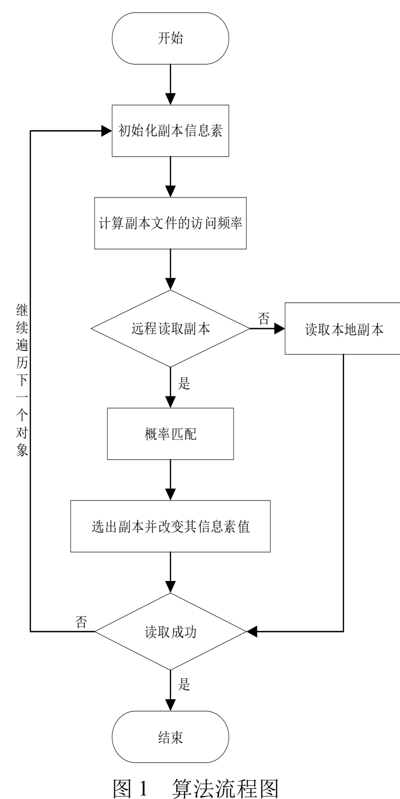


图 1 算法流程图

Fig. 1 Algorithm flow chart

3 实验设计与结果分析

本实验采用 CloudSim 云计算仿真器进行实验的仿真。CloudSim 支持云计算环境的资源模拟，可以对副本管理中相关环境进行仿真模拟。利用 CloudSim 的扩展性，修改 Host 和 DataCenter 类，添加新的副本放置组件完成副本放置工作，比较副本访问时间的开销、网络利用率和负载均衡度。本实验对改进蚁群算法、原始蚁群算法和 Min-Min 算法进行仿真。实验环境为：软件采用 window10 操作系统 CloudSim3.0.3，硬件采用 i5-7300HQ@2.5GHz，内存为 8GB。

CloudSim 仿真实验步骤：

a)初始化 CloudSim 包。

b)创建数据中心。

c)创建数据代理中心。

d)创建云任务。

e)读取副本，并进行副本对象的放置。

f)启动仿真。

g)仿真结束后分析结果。

本实验中算法设计的参数如表 1 所示。

表 1 蚁群算法基本数据

Table 1 Basic Data of Ant Colony Algorithms

名称	值
作业数	100~800
启发因子 α	3
期望启发因子 β	1.5
信息素挥发因子 ρ	0.2~0.8
蚂蚁数量 m	50
最大迭代次数 n	200

本次实验分为三组，分别从平均作业完成时间、网络利用率、负载均衡度三个方面对改进蚁群算法、原始蚁群算法、Min-Min 算法进行性能评估，并对实验结果进行比较。最终形成本次仿真实验。

1）平均作业完成时间

为了验证改进蚁群算法在作业完成时间的优势，本文选择与原始蚁群算法和 Min-Min 算法进行对比，为了准确地对比三种不同的算法，引入平均作业完成时间。

$$\bar{T}=\sum_{i=1}^{n}T_i\Big/n \quad (9)$$

其中：T_i 为第 i 个作业的运行时间，n 为运行作业的总数。平均作业时间是指执行所有作业的整个时间与作业完成总数的比值。

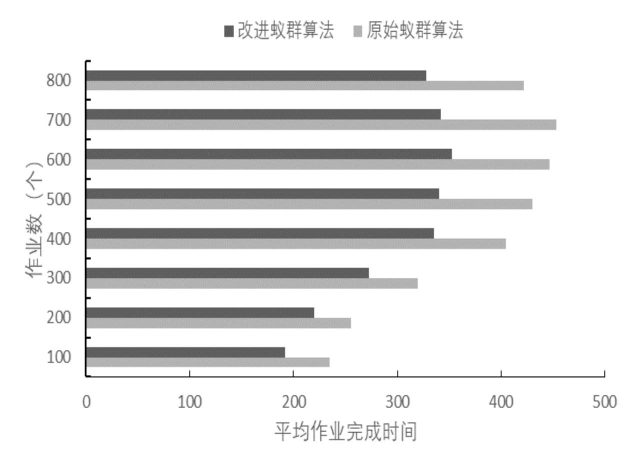


图 2 平均作业完成时间

Fig. 2 Average homework completion time

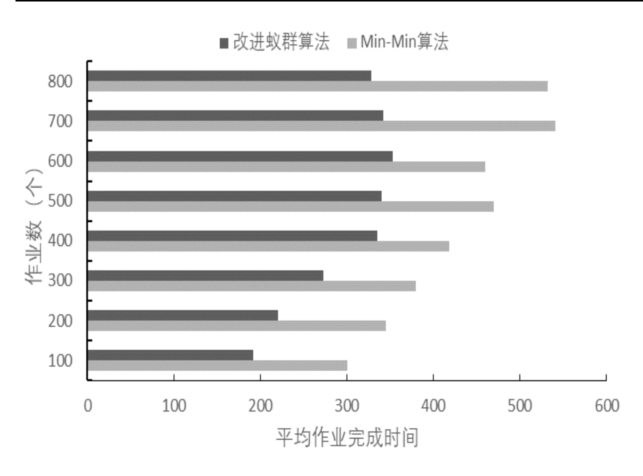


图 3　平均作业完成时间

Fig. 3　Average homework completion time

通过图 2 和 3 实验数据分析可得,在作业完成时间上，改进的蚁群算法比原始蚁群算法和 Min-Min 算法都要短。因为 Min-Min 算法总是将作业分配到处理时间最短的资源上，使一部分资源处于一直处于工作状态，而另外一部分一直处于空闲状态，以保证作业能够最短时间完成，这样就造成作业分配不合理。而蚁群算法的全局搜索能力使其在作业完成时间上更具有优势。改进之后的蚁群算法，在进行信息素和概率选择的优化后，与原始蚁群算法相比，作业完成的效率上有明显的改善。从图中可以看出改进的蚁群算法对于原始蚁群算法平均作业完成时间减少了 14%~25%。与 Min-Min 算法相比，平均作业完成时间减少了 27%~40%。

2）网络的利用率

为了描述副本放置时需要更新副本，并由此而需要消耗时间和占用网络带宽，定义网络利用率来表示不同算法对网络的利用程度。

$$R_{ENU}=\frac{N_{remote-file}+N_{file-replicas}}{N_{local-file}} \tag{10}$$

其中：$N_{remote-file}$ 是指计算单元从远程节点访问副本的次数，$N_{file-replicas}$ 是指作业运行放置时相关的副本数，$N_{local-file}$ 是指从本地节点中访问副本的次数。R_{ENU} 越小，表示算法的网络利用率小，同时该算法占用的网络资源就小，给网络带来的负载就小，相应也会说明该算法表现更优秀。

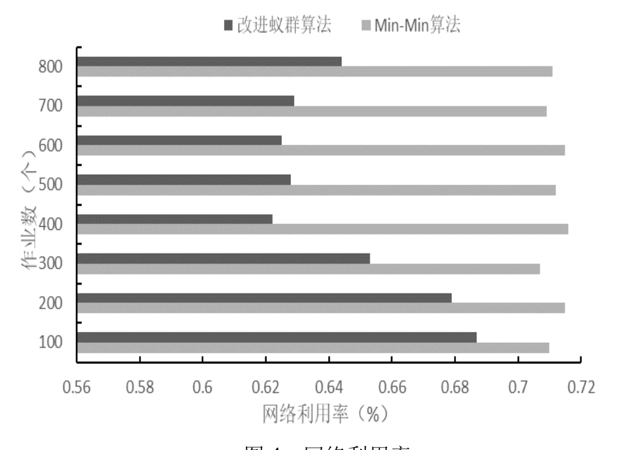


图 4　网络利用率

Fig. 4　Network utilization ratio

从图 4 中可以看出改进之后的蚁群算法相比于 Min-Min 算法差别很大，根据前面给出的定义和式（10），改进之后的蚁群算法在网络负载方面表现更加优异，相比 Min-Min 算法，提升约 8.5 个百分点。

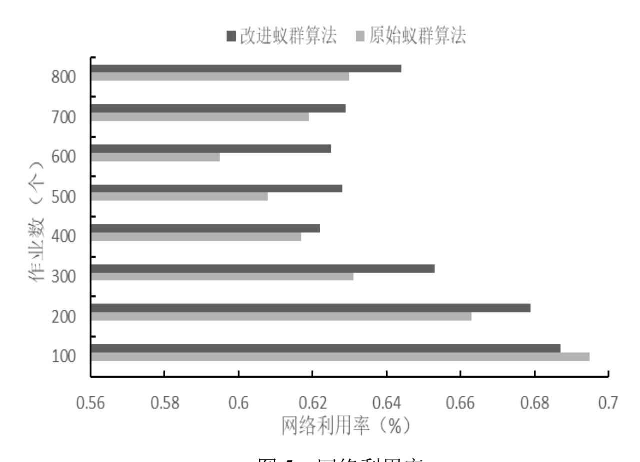


图 5　网络利用率

Fig. 5　Network utilization ratio

从图 5 中可以看到，虽然改进之后蚁群算法在不同作业数下网络利用率数值波动较大。但从整体上看，对比原始蚁群算法，改进之后的蚁群算法在网络利用率上表现更好，提升了约 0.2 个百分点。

3）负载均衡度

在实验的过程中，在不同的作业数的情况下，对负载均衡程度进行了统计。计算分析可以得出不同算法在作业数增加时，负载均衡的情况。负载均衡度 *LB* 的公式为

$$LB=\sqrt{\frac{1}{m}\sum_{j=1}^{m}(c_j-\bar{c})^{2}} \tag{11}$$

从图 6 中可以看出，改进蚁群算法相对于 Min-Min 算法相比，由于系统资源节点作出实时的调整，故改进蚁群算法有较大的优势。相对于原始蚁群算法，改进之后的蚁群算法表现得略好。

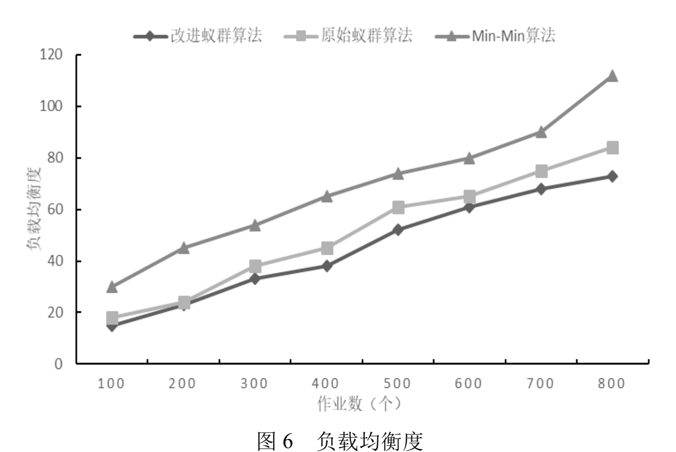


图 6　负载均衡度

Fig. 6　Load balancing degree

4　结束语

针对云环境下多副本管理中的副本放置问题，本文提出了一种基于改进蚁群优化的副本放置算法。通过蚁群算法的信息动态更新方式和改进概率选择方式使蚁群能够更好的进行路径的选择，最终把副本放置在最佳的位置。全面考虑副本放置性能对调度效率和访问性能的影响，并在云计算模拟器 CloudSim 中进行了模拟实验。用三组实验对本文改进的算法进行验证，实验结果表明与原始蚁群算法、Min-Min 算法相比，改进之后的蚁群算法在平均作业完成时间上有着优秀的表现，能够有效地减少副本放置过程时间的消耗，同时能更好的平衡网络的负载。在接下来的工作中，研究副本创建和副本定位等副本策略其他领域将是下一步的工作重心。

参考文献：

[1] 张翠苹，郭振洲，拱长青．云存储环境下副本选择策略研究 [J]. 计算机科学, 2015, 42 (S2): 408-412. (Zhang Cuiping, Guo Zhenzhou. Gong Changqing. Study on strategy of replica selection in cloud storage environment [J]. Computer Science, 2015, 42 (S2): 408-412.)

[2] 左方，何欣．一种基于蚁群算法的云存储副本动态选择机制研究 [J]. 计算机应用研究, 2015,32 (11): 3368-3370. (Zuo Fang, He Xin, Pherom-one-based ant colony replica selection mechanism in cloud storage [J]. Application Research of Computers, 2015,32 (11): 3368-3370.)

[3] 吴修国．云计算环境下面向最小成本的数据副本策略 [J]. 计算机科学, 2014, 41 (10): 154-159. (Wu Xiuguo. Minimum-cost based data rep-lication strategy in cloud computing environment [J]. Computer Science, 2014, 41 (10): 154-159.)

[4] 祝家钰，肖丹．云计算架构下的动态副本管理策略 [J]. 计算机工程与设计, 2012, 33 (9): 3362-3365. (Zhu Jiayu, Xiao Dan. Dynamic replicati-on management scheme for cloud computing [J]. Computer Engineering & Design, 2012, 33 (9): 3362-3365.)

[5] 李君，侯孟书．基于萤火虫优化的副本放置方法 [J]. 计算机应用研究, 2018, 36 (3): 314-316. (Li Jun, Hou Mengshu. Replica placement strateg-y based on glowworm swarm optimization [J]. Application Research of Computer, 2018, 36 (3): 314-316.)

[6] 陶永才，张宁宁，石磊，等．异构环境下云计算数据副本动态管理研究 [J]. 小型微型计算机系统, 2013, 34 (7): 1487-1492. (Tao Yongcai, Zhang Ningning, Shi Lei, *et al.* Research on dynamic management of data replicas of cloud computing in hetergeneous environments [J]. Journal of Chinese Computer Systems, 2013, 34 (7): 1487-1492.)

[7] 孙新，李庆洲，赵璞，等．对等网络中一种优化的副本分布方法 [J]. 计算机学报, 2014, 37 (6): 1424-1434. (Sun Xin, Li Qingzhou, Zhao Pu, *et al.* An optimize replicadistribution method for peer-to-peer netw-ork [J]. Chinese Journal of Computers [J] , 2014, 37 (6): 1424-1434.)

[8] Sahoo J, Glitho R. Greedy heuristic for replica server placement inCloud based content delivery networks [C]//Proc of IEEE Symposium on Computers and Communication.Washington DC:IEEE Computer Society, 2016: 302-309.

[9] 饶磊，杨凡德，李新明，等．基于热度分析的动态副本创建算法 [J]. 计算机应用，2014，34 (S2): 130-134. (Rao Lei, Yang Fande, Li XinMing, Liu Dong, *et al.* Dynamic replica creation algorithm based on tempera-ture ' s analysis [J]. Journal of Computer Applications, 2014 ,34(S2): 130-134.)

[10] Rajalakshmi A, Vijayakumar D, Srinivasagan K G. An improve-d dynamic data replica selection and placement in cloud [C]// Proc of International Conference on Recent Trends in Information Technology. 2014.

[11] 刘田甜，李超，胡庆成，等．云环境下多副本管理综述 [J]. 计算机研究与发展, 2011, 48 (S3): 254-260. (Liu Tiantian, Li Chao, Hu Qing, *et al.* Multiple replicas management in the cloud environment [J]. Journal of Computer Research and Development, 2011, 48 (S3): 254-260.)

[12] 黄冬梅，杜艳玲，贺琪，等．基于多属性最优化的海洋监测数据副本布局策略 [J]. 计算机科学, 2018, 45 (6): 72-75,104. (Huang Dongmei, Du Yanling, *et al.* Marine monitering data replica layout strategy based on multiple attribute optimization [J]. Computer Science, 2018, 45 (6): 72-75,104.)

[13] Singh A, Thapar S, Bhatia A, *et al.* Disk scheduling using a customized discrete firefly algorithm [J]. Cogent Engineering, 2015, 2 (1): 1011929.

[14] 赵秋云．基于相似场境推荐的数据网格副本选择策略 [J]. 微电子学与计算机, 2012, 29 (9): 23-26+30. (Zhao Qiuyun. Replica Selection strate-gy based on similar scene recommendation in data grid [J]. Microelectronics & Computer, 2012, 29 (9): 23-26+30.)

[15] Long S Q, Zhao Y L, Chen W. MORM: a multi-objective optimized replication management strategy for cloud storage cluster [J]. Journal of Systems Architecture, 2014, 60 (2): 234-244.